International Journal of Future Generation Communication and Networking
Vol. 13, No. 4, (2020), pp. 2684 - 2693

An Optimized Data Duplication Strategy for Federated Clouds using Bloom Filters

Nipun Chhabra[a*], Manju Bala[b]
[a]*Ph.D. Research Scholar, I.K.Gujral. Punjab Technical University, Kapurthala (Punjab), India.*
[b]*Director, Khalsa College of Engineering & Technology, Amritsar , (Punjab), India.*
[a]*nipunchhabra5@gmail.com,* [b]*drmanju571@gmail.com*

Abstract—In Cloud computing deduplication falls under IAAS model as it directly deals with the storage in the datacenters. Deduplication has been implemented at various levels such as file level, block level and chunk level as per the organizational requirements. In the due course of time, many companies have offered for providing better services and subsequently interoperability has braught many Cloud Service Providers (CSP) for scalable provisioning of services under variable workloads, resources and network conditions as hybrid clouds. This concept was very beneficial for customers as well as CSPs and hence it has further evolved and brought many CSPs, either private or public, under one group/alliance and referred it as Federated cloud. Despite of vast storage offered in federated cloud structure, much of it is exhausted with redundant copies of data which needs to be made distinct. In this paper, we have rendered an optimized deduplication strategy for federated cloud environment in which the domain of the storage in not limited to just one CSP but all the CSPs under one federation. It has been observed that the work pertaining to this is not available and in this optimized deduplication strategy we have used file level deduplication technique for the implementation of this approach.

Keywords: *Optimized Deduplication Strategy using Bloom Filters, Deduplication in Federated Cloud, Interoperability in Cloud, Probabilistic Data Structure, Bloom Filters, Deduplication, Cloud Computing, Cloud Data Storage, Storage Optimization in Cloud Computing.*

1. Introduction

Due to reduced cost, scalability and ease of access, cloud computing is gaining popularity day by day. Cloud Computing allows organizations to expand or reduce their computing facilities very quickly, without upfront costs of setting up an IT infrastructure [1]. Considering the benefits of cloud computing many people/organizations are moving/storing their data on cloud but the inflow of the bulks of data is such that even vast cloud data storage servers are falling short to accommodate the pace. For coping up with the needs, different cloud providers are coming together to meet the increased requirements of users which can be storage, computing or other resources through interoperability[2]. As the cloud computing is a real time scenario and with the advancements in the internetworking technology more and more netizens are joining in the cloud services resulting in the extra load on CSPs. Interoperability has presented a solution to load shedding between the CSPs by bringing them under one umbrella as a federation which are bind together by Service Level Agreements (SLA). To achieve interoperability, cloud service providers (CSP) adopt/employ standard interfaces, formats, architectural components and protocols that facilitate the collaboration amongst various clouds. Interoperation can be of Hybrid Clouds, Multi Clouds or Federated Clouds but when transparent interoperation is enabled between cloud providers through direct agreements it is known as cloud federation.

Cloud federation allows users to share their resources through federation policies and if one of the service providers is unable to meet the customer requirements, then that request can be outsourced to other service providers of that federation. Moreover, cloud federation allows providers operating at low utilization to lease part of their resources to other federation members in order to avoid wasting their non-storable compute resources. It has been observed that despite of the increased storage with federation, much of the cloud storage is occupied by duplicate copies of data which needs to be filtered to optimize the storage capacity. In the next section a detailed literature survey has been carried out on the evolution of federation in cloud computing and possibilities of conducting deduplication in cloud environment.

2. Literature Survey

Numerous researchers have been studying the concept of Interoperability to bring multiple clouds under one alliance/federation. In this section evolution of federation in cloud has been covered.

2.1 Evoultion of Federation in Cloud

[3] presented the concept of Intercloud, which is an extension of computing and Storage capacity provided to a single cloud through interconnected global "cloud of clouds". Various issues of Intercloud topology have been deliberated by [4] in detail about Intercloud Collaboration, Intercloud Security and Intercloud Resources Directory to render a better design of the system. [5] proposed that many service providers bound by standard SLAs come together to form a federation where each service provider serves the client based on the end resources. [6] presented an innovative approach where various cloud providers are connected with each other to deliver a universal decentralized computing system based on open standards where, this union is lead by agreements and constraints in the interconnected web of infrastructure. For achieving QoS under variable workload in Federated cloud computing environment [7] proposed an approach which handles provisioning of application services in a scalable computing resources and network conditions.
[8] used cloudsim as a framework to model a solution based on three pillars viz. Cloud Exchange, Cloud Brokers and Cloud coordinators to simulate Cloud Computing Services and Infrastructure and it has shown very promising outcomes. For establishing highly dynamic cloud federations [9] proposed the concept of Dynamic Cloud Collaboration (DCC) where, the role of primary cloud provider (pCP) is taken up by one cloud provider (CP) and other participant/ federated cloud providers are called collaborating CPs. For adding up more CPs/resources to a DCC Platform, permission of other CPs based on their agreement and policies is also desired.

[10] presented an analytical model that characterizes Cloud federation and can be used to drive provider's decisions about resource outsourcing, insourcing, and node shutdown also. Authors have also studied the effect of these decisions on the provider's profit and evaluated the most appropriate provider's configuration depending on the environment conditions. [11] presented a platform which allows the user/customers to choose from a central platform of federated clouds. Here the customers get more options and price policies available in the federation of heterogeneous Cloud providers. [12] deliberated that in cloud federations CSPs participate willingly by agreeing on an Federation Level Agreement (FLA) on the other hand intercloud organizations have no such constraints and policy agreements for private or public cloud for participation. Moreover, cloud federations have a broker for communication between CSPs while no such agent is there in intercloud and they follow open standards for interoperability. [13] advocated that Cloud Federation is the collection of many autonomous services/resources behaving like one entity under coalition. To make cloud computing practically possible federation of assets, resources and services should also be brought under one umbrella. [14] researched on MetaStorage in federated environment where Distributed hash tables were used to index data on different storage entities.

The main agenda behind the research field of Cloud Federation is to gain good QoS, cost efficiency, reliability, effective utilization of cloud and to achieve excellent performance. Next subsection discusses the concept of deduplication its techniques and implementation in cloud environment.

2.2. Data Deduplication in Cloud

More and more organizations are using cloud computing for processing and storing their data[15]. Out of the many valuable services being offered in cloud computing, cloud storage is the most exhausted upon service. Organizations are buying more storage space and employing more efficiency in managing the stored data including networking and backups. Even the regular backups leads to the replication of data. Moreover, the identical files created within an organization to work upon independently, by different departments of the organization, leading to redundancy of data and wastage of valuable storage space. The only solution to this problem is deduplication, commonly known as single instance storage. In this technique a single copy of data is stored and a logical pointer is created for this data so that users can easily access the data through pointer as and when required in future. Deduplication techniques are categorised on the basis of granularity, location and time. In granularity method it is checked that whether deduplication is done at file level or block level (where chunks can be of fixed/variable length). Secondly, location based deduplication can be source-side (before uploading the data) or destination-side (at the storage end) and if these techniques are implemented in reference to time, deduplication can be Inline or post-process[16].

As, several users are placing multiple copies of their data on the cloud regularly and to accommodate them all more storage space is needed, deduplication proves to be a very effective and efficient technique for conserving/harvesting valuable cloud storage. In deduplication only unique copy of data is stored and all other

copies are removed and it can be performed keeping in view the privacy and confidentiality of data using various cryptographic solutions and involving symmetric encryption [17] [18] [19]. In some cases the use of threshold secret sharing is also proposed while maintaining the robustness of key management [20] [21].

[22] proposed message-locked encryption for space-efficient secure outsourced storage. Convergent encryption leads to a significant number of convergent keys. [23] proposed a novel technique and named it as Dekey which is a convergent key management scheme used for secure deduplication. Dekey proves to be a very efficient and reliable scheme which segregates and shares convergent key across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data.
A secret sharing method, for data confidentiality has been proposed by [24] which is used to encode fragmented data, instead of encryption mechanisms. The data's secret shares will be accessible by only those authorized users who own the corresponding data copy. Burton Howard has proposed an advanced technique and named it as Bloom filter which is a probabilistic data structure coined for look up operations. This scheme proves to be space and time efficient to test for the membership of an element in a set. Sometimes it returns false positives also but it promises to rule out any false negatives [25].
From the above study, it has been observed that with the increasing demand of computing and storage resources on cloud, more CSPs are coming together to form a federation to cater to the needs of users. It has been observed that users are storing multiple copies of data on cloud storage in federated cloud environment resulting into wastage of precious memory which can be easily optimised using deduplication in federated cloud environment.
The rest of the paper has been arranged as follows; section 3 presents the research gap and formulates the problem statement from existing system and elucidates the design of an optimised deduplication strategy for federated cloud environment. Section 4 records the observations, section 5 deliberates results and discussions and section 6 concludes the study in the paper and confers the future scope of this work.

3. Research Gap and Proposed System

From deep analysis of the technical literature it was observed that in the prevailing systems the concept of deduplication has been implemented mostly on public and private cloud deployment models and in some cases, to meet the demands of the users/organisations secure deduplication in hybrid clouds is also researched.
It was further observed that in the existing system hashing technique has been used for searching duplicate copies which is relatively slow and a faster technique can be deployed to optimize the performance.

With the advancement of the technology many CSPs, whether public or private, came under one umbrella for load sharing and form a federation of CSPs called as Federated cloud which is bound under common Federation Level Agreements (FLA). In federated CSPs the search of duplicate copies is not limited to one CSP but it is searched in the group of public or private CSPs participating in the Federation.

Careful study of technical literature confers that there is not much work done in the area of deduplication in Federated clouds so there is a huge research gap and enough scope for rendering/designing an optimised deduplication strategy for federated clouds.

3.1. Existing System

In the existing system the concept of deduplication has been implemented in private and public cloud environment using hashing technique for searching the duplicate copies along with simple encryption technique, for securing the data during this process. Here, the data/file is first encrypted on the client side and then at the server side hashing technique is implemented for deduplication of data on cloud storage, as explained below.

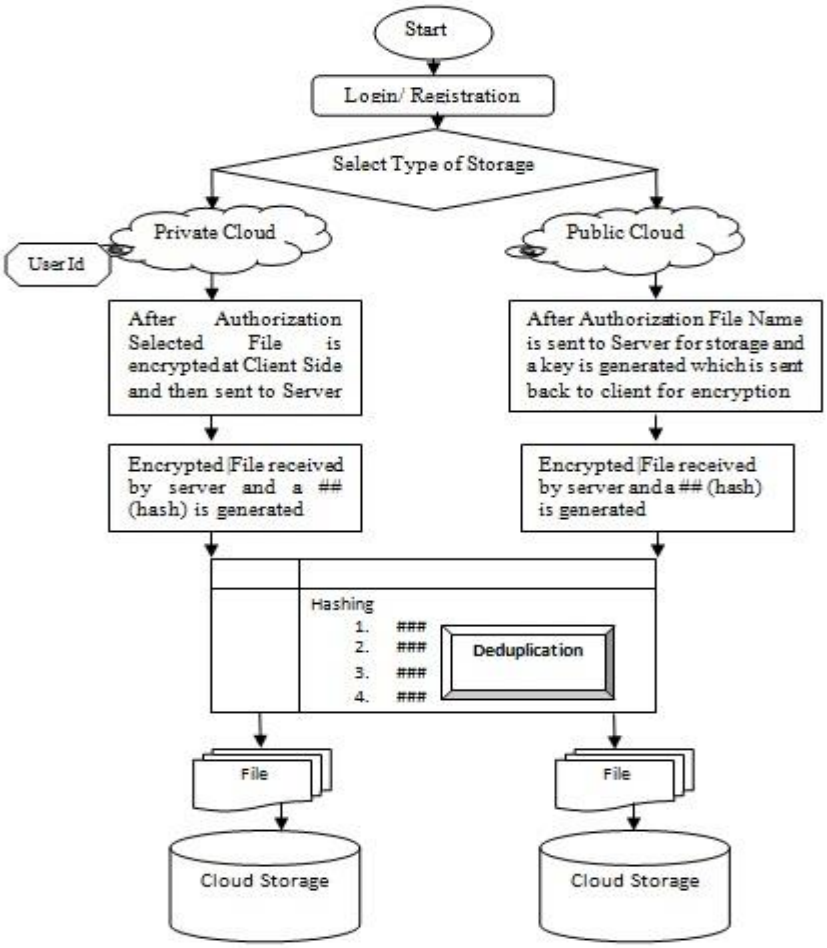

Fig. 1. Diagrammatic representation of the working of the Existing System

3.2. Problem Formulation/Statement

From the deep survey of technical literature and analysing the existing system thoroughly, it was found that an optimized deduplication strategy can be designed by incorporating bloom filters for deduplication in federated cloud environment rather than implementing on private/public cloud system. Bloom filters are not only help in indexing the file but also help in searching the desired file in lesser span of time than hashing. Design of the proposed system is discussed in next section given ahead.

3.3. Proposed System

To overcome the shortcomings of the existing system and to optimize the deduplication in cloud system the proposed strategy has been implemented in a federated cloud environment to incorporate interoperability in participating CSPs. The proposed system has been implemented in the simulated environment of cloudsim 3.0.3 using framework of java.
In the proposed Federated system design, two private CSPs and three public CSPs have been brought under one alliance/federation to simulate the federated cloud environment. User selects the type of CSP, whether private or public, and the selected file is encrypted with users' credentials in former case otherwise the selected file name is sent to the server which generates the key and sends back to the client for file encryption. The next step is checking the duplicate copy of the file on selected CSP with Look up operation using Bloom filters

3.4. Bloom Filters

Bloom filters are space and time efficient probabilistic data structures which are deployed to check the membership of an element in a dataset in relatively lesser time than hashing. The presence of any element during lookup operation is depicted with one and the absence is depicted with 0. Bloom filters make use of only

3 or 4-bytes which is far smaller than the traditional hash codes which are 16 to 64 bytes long. Therefore it was decided to deploy bloom filters for lookup operations in deduplication technique.

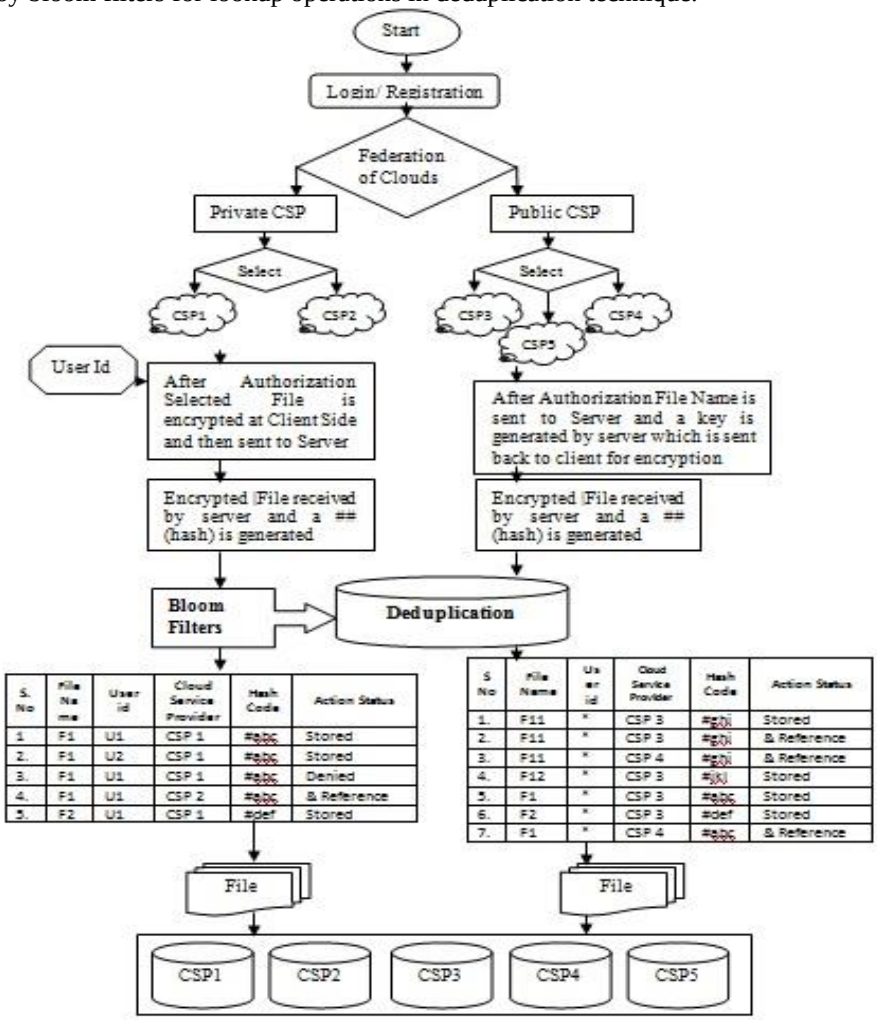

Fig. 2. Diagrammatic representation of the working of the Proposed System

3.5. Methodology

In the present study have proposed an optimised data duplication strategy for federated cloud environment using Bloom Filters and the sequence of events in the proposed system are as follows:

1. CloudSim 3.03 package is initialized by creating the Datacenter Broker, virtual machines and cloudlets.
2. Private access: After authorized login by user in chosen Private CSP the selected file is encrypted with users' credentials and sent to the server where a hash code of encrypted file is generated by the server.
3. Bloom Filter: Look up operation is conducted against generated hash
4. Deduplication: Checked for existence of file on the storage against UserId and hash code of file to be unique
5. A reference is passed if the file exists otherwise the new file is stored
6. Public access: After authorized login by user in chosen Public CSP, the selected file name is sent to the server which generates the key and sends back to the client.
7. The selected file is encrypted through this key and sent to the server where a hash code is generated for this encrypted file.
8. Bloom Filter: Look up operation is conducted against generated hash
9. Deduplication: Checked for duplicate file on the storage against hash code of file.
10. A reference is passed if the file exists otherwise the new file is stored.

4. Observations

Experimental Results or the simulated observations of the Proposed system are mentioned in Table 1 & Table 2 as given below:

Table 1. Actions performed during deduplication in federated clouds in Private CSPs

S. No.	File Name	User id	Cloud Service Provider	Hash Code	Action Status
1	F1	U1	CSP 1	#abc	Stored
2.	F1	U2	CSP 1	#abc	Stored
3.	F1	U1	CSP 1	#abc	Denied
4.	F1	U1	CSP 2	#abc	& Reference
5.	F2	U1	CSP 1	#def	Stored

Case I: Actions performed during deduplication in federated clouds in private CSPs/cloud storage as Shown in the Table 1

i. At searial no 1, when a user U1 tries to upload a file F1 in Cloud Service Provider CSP1 for the first time, the file F1 gets stored and action status is updated to 'Stored'.
ii. At serial. no. 2, when a different user U2 sends a request to upload the same file F1 in CSP1, the file F1 also gets stored and action status is updated to 'Stored'.
iii. At serial no. 3 when a user U1 sends a request to upload a file F1 in CSP1 the file F1 is found on the same CSP (CSP1) during look up operation (using Bloom Filters) with same hash code (#abc) and the action status is updated to 'Denied'.
iv. At serial no. 4 when a user U1 sends a request to upload a file F1 in CSP2 the file F1 is found on CSP1 during look up operation (using Bloom Filters) with same hash code (#abc) then the reference to file F1 is passed instead of saving the same file again and the action status is updated to '& Reference.
v. At serial no. 5 when a user U1 sends a request to upload a file F2 in CSP1 the file F1 is Stored on CSP1 and the action status is updated to 'Stored'

Table 2. Actions performed during deduplication in federated clouds in Public CSPs

S. No.	File Name	User id	Cloud Service Provider	Hash Code	Action Status
1.	F51	PU	CSP 3	#ghi	Stored
2.	F51	PU	CSP 3	#ghi	& Reference
3.	F51	PU	CSP 4	#ghi	& Reference
4.	F52	PU	CSP 3	#jkl	Stored
5.	F1	PU	CSP 3	#abc	Stored
6.	F2	PU	CSP 3	#def	Stored
7.	F1	PU	CSP 4	#abc	& Reference

In Public cloud storage, user id of users interested in data storage are not maintained in the table.

Case II: Actions performed during deduplication in federated clouds in Public CSPs/Cloud Storage as shown in the Table 1.2

i. At searial no 1, in case of public CSPs, when a public user PU tries to upload a file F51 in Cloud Service Provider CSP3 for the first time, the file F51 gets stored and action status is updated to 'Stored'.
ii. At serial no. 2 when a public user PU sends a request to upload a file F51 on CSP3, the file F51 is found on CSP3 during look up operation with same hash code (#ghi) then the reference to file F51 is passed instead of saving the same file again and the action status is updated to '& Reference.

iii. At serial no. 3 when a public user PU sends a request to upload a file F51 on CSP4, the file F51 is found on CSP3 during look up operation (using Bloom Filters) with same hash code (#ghi) then the reference to file F51 is passed instead of saving the same file again and the action status is updated to '& Reference.

iv. At searial no 4, when a public user PU tries to upload a file F52 in Cloud Service Provider CSP3 for the first time, the file F52 gets stored and action status is updated to 'Stored'.

v. At searial no 5, when a public user PU tries to upload a file F1 in Cloud Service Provider CSP3 for the first time, the file F1 gets stored and action status is updated to 'Stored'. As the file F1 is the same file which is stored at Private Cloud Storage CSP1 and it is being stored at public CSP3, again.

vi. At searial no 6, when a public user PU tries to upload a file F2 in Cloud Service Provider CSP3 for the first time, the file F2 gets stored and action status is updated to 'Stored'. As the file F2 is the same file which is stored at Private Cloud Storage CSP1 by the same user U1 and it is being stored at public CSP3, again.

vii. At searial no 7, when a public user PU tries to upload a file F1 in Cloud Service Provider CSP4, the file F1 is found on CSP3 during look up operation (using Bloom Filters) with same hash code (#abc) then the reference to file F1 is passed instead of saving the same file again and the action status is updated to '& Reference'

5. Results and Discussions

Above mentioned simulation results are obtained on the basis of following assumptions:

a) Private CSPs can store same file from same users atleast once and in public CSPs the same file can be stored once as it is available in public domain for other to mark the reference.
b) In case the user tries to delete the original file in Public CSP then instead of deleting the File, the reference (address) to the file is deleted for that User.

5.1. Memory Optimisation.

The proposed strategy is proving very beneficial in harvesting the memory from Cloud Storage (duplicated data) as shown in the Table 3 for Private CSPs & Table 4 for Public CSPs as given below:

Table 3. Harvesting the memory from Private CSPs using proposed System

S.No	File Name	File Size	User id	Cloud Service Provider	Action	Memory used after Deduplication
1	F1	100kb	U1	CSP 1	Stored	100kb
2	F1	100kb	U2	CSP 1	Stored	100kb
3	F1	100kb	U1	CSP 1	Denied	-
4	F1	100kb	U1	CSP 2	& Reference	-
5	F2	200kb	U1	CSP 1	Stored	200kb

As shown in the Table 3, till the time deduplication has not been implemented the memory requirement was 600 kbs but after deploying optimised data duplication strategy for federated cloud environment using Bloom Filters the memory required for file storage has reduced to 400 kbs and 200kbs of memeory is saved.

Table 4. Harvesting the memory from Public CSPs using proposed System

Sr. No.	File Name	File Size	Cloud Service Provider	Action Status	Memory used after Deduplication
1.	F11	200kb	CSP 3	Stored	200kb
2.	F11	200kb	CSP 3	& Reference	-
3.	F11	200kb	CSP 4	& Reference	-
4.	F12	300kb	CSP 3	Stored	300kb
5.	F1	100kb	CSP 3	Stored	100kb

6.	F2	200kb	CSP 3	Stored	200kb
7.	F1	100kb	CSP 4	& Reference	-

As shown in the Table 4 for public CSPs, after applying the proposed strategy the memory requirement was reduced to 800 kbs which was otherwise 1300kbs.

5.2. Time optimization in Deduplication during Look Up operation using Bloom Filters

It was observed that in deduplication by using bloom filters for the file look up operations, the time taken was much lesser as compared to the look up operations with hashing techniques and the same can be seen in the Table 5 where simulated results for the existing sytem and proposed system have been recorded.

Table 5. Time taken for Look Up Operation in Existing sytem & Proposed Ssytem

	Existing System		**Proposed system**	
Size	**Private**	**Public**	**Private**	**Public**
100	78	79	46	63
200	62	85	46	32
300	93	50	62	36
400	62	45	47	15
500	62	86	47	31
600	47	78	31	46
700	63	78	47	47

From the Fig 3, Fig 4, Table 3 & Table 4 it is evident that the proposed system i.e. optimised data duplication strategy for federated cloud environment using Bloom Filters shows excellent results and saves memory as well as time with the proposed solution as compared to the existing system.

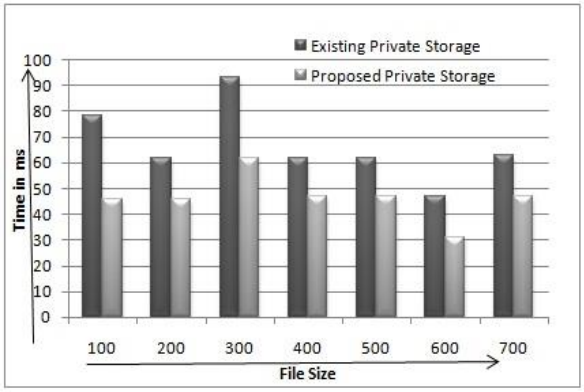

Fig 3. 3-D Cluster-Column Graphical Representation of the Simulation Results for Look Up Time for Private Storage

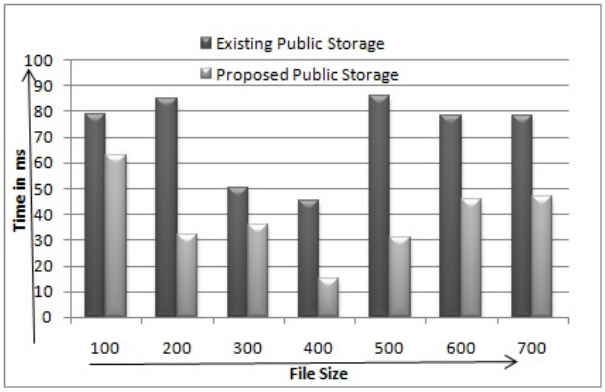

Fig 4. 3-D Cluster-Column Graphical Representation of the Simulation Results for Look Up Time for Public Storage

6. Conclusion

To overcome the meagrenss of the existing system and to reduce data duplication in cloud environment, an optimised data duplication strategy has been rendered for federated cloud environment using Bloom Filters. CloudSim 3.0.3, a Cloud Simulation Tool has been used to implement and test the proposed algorithm using the framework of Java. The obtained results were compared against the existing system and results indicate that the proposed system shows excellent results by harvesting memory and reduces time with the proposed solution as compared to the existing system. In future, this technique can be further advanced by presenting flexible & friendly FLAs so that more CSPs can collaborate and offer services. More research can be done on cental management of the federation through a common agent/broker.

References

[1] Sharma, V., & Bala, M. (2020). An Improved Task Allocation Strategy in Cloud using Modified K-means Clustering Technique. Egyptian Informatics Journal. doi:10.1016/j.eij.2020.02.001

[2] Ranjan, R. (2014). *The Cloud Interoperability Challenge. IEEE Cloud Computing, 1(2), 20–24.* doi:10.1109/mcc.2014.41

[3] Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., & Morrow, M. (2009). *Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability. 2009 Fourth International Conference on Internet and Web Applications and Services.* doi:10.1109/iciw.2009.55

[4] Bernstein, D., D. Vij, "*Intercloud Exchanges and Roots Topology and Trust Blueprint*", Huawei Technologies, Ltd, Santa Clara, California, USA

[5] Suma, V., Bouhmala, N., & Wang, H. (Eds.). (2021). *Evolutionary Computing and Mobile Sustainable Networks. Lecture Notes on Data Engineering and Communications Technologies.* doi:10.1007/978-981-15-5258-8

[6] Kertesz, A. (2014). *Characterizing Cloud Federation Approaches. Cloud Computing, 277–296.* doi:10.1007/978-3-319-10530-7_12

[7] Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). *InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. Lecture Notes in Computer Science, 13–31.* doi:10.1007/978-3-642-13119-6_2

[8] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2010). *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23–50.* doi:10.1002/spe.995

[9] Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010). *How to Enhance Cloud Architectures to Enable Cross-Federation. 2010 IEEE 3rd International Conference on Cloud Computing.* doi:10.1109/cloud.2010.46

[10] Goiri, I., Guitart, J., & Torres, J. (2010). *Characterizing Cloud Federation for Enhancing Providers' Profit. 2010 IEEE 3rd International Conference on Cloud Computing.* doi:10.1109/cloud.2010.32

[11] Zangara, G., Terrana, D., Corso, P. P., Ughetti, M., & Montalbano, G. (2015). *A Cloud Federation Architecture. 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC).* doi:10.1109/3pgcic.2015.183

[12] Kogias, D. G., Xevgenis, M. G., & Patrikakis, C. Z. (2016). *Cloud Federation and the Evolution of Cloud Computing. Computer, 49(11), 96–99.* doi:10.1109/mc.2016.344

[13] https://auth0.com/learn/cloud-identity-access-management/

[14] Bermbach, D., Klems, M., Tai, S., & Menzel, M. (2011). *MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs. 2011 IEEE 4th International Conference on Cloud Computing.* doi:10.1109/cloud.2011.62

[15] Sharma, V., & Bala, D. M. (2018). A Credits Based Scheduling Algorithm with K-means Clustering. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). doi:10.1109/icsccc.2018.8703201

[16] Chhabra, N., & Bala, M. (2018). *A Comparative Study of Data Deduplication Strategies. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC).* doi:10.1109/icsccc.2018.8703363

[17] Wang, W., Li, Z., Owens, R., & Bhargava, B. (2009). *Secure and efficient access to outsourced data. Proceedings of the 2009 ACM Workshop on Cloud Computing Security - CCSW '09.* doi:10.1145/1655008.1655016

[18] Kamara, S., & Lauter, K. (2010). *Cryptographic Cloud Storage. Lecture Notes in Computer Science, 136–149.* doi:10.1007/978-3-642-14992-4_13

[19] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2011). *Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. IEEE Transactions on Parallel and Distributed Systems, 22(5), 847–859.* doi:10.1109/tpds.2010.183

[20] Geambasu, R., Kohno, T., Levy, A.A, Levy, M.H. (2009). *Vanish: Increasing Data Privacy with Self-Destructing Data.* Conference: 18th USENIX Security Symposium, Montreal, Canada, August 10-14, 2009, Proceedings

[21] Tang, Y., Lee, P. P. C., Lui, J. C. S., & Perlman, R. (2012). *Secure Overlay Cloud Storage with Access Control and Assured Deletion. IEEE Transactions on Dependable and Secure Computing, 9(6), 903–916.* doi:10.1109/tdsc.2012.49

[22] Bellare, M., Keelveedhi, S., & Ristenpart, T. (2013). *Message-Locked Encryption and Secure Deduplication. Lecture Notes in Computer Science, 296–312.* doi:10.1007/978-3-642-38348-9_18

[23] Li, J., Chen, X., Li, M., Li, J., Lee, P. P. C., & Lou, W. (2014). *Secure Deduplication with Efficient and Reliable Convergent Key Management. IEEE Transactions on Parallel and Distributed Systems, 25(6), 1615–1625.* doi:10.1109/tpds.2013.284

[24] Li, J., Chen, X., Huang, X., Tang, S., Xiang, Y., Hassan, M. M., & Alelaiwi, A. (2015). *Secure Distributed Deduplication Systems with Improved Reliability. IEEE Transactions on Computers, 64(12), 3569–3579.* doi:10.1109/tc.2015.2401017

[25] B. H. Bloom, "Space/time trade-o s in hash coding with allowable errors," Comm. of the ACM, vol. 13, no. 7, pp. 422–426, 1970.